International Journal of Electrical and Computer Engineering (IJECE)
Vol. 15, No. 6, December 2025, pp. 5986~6000
ISSN: 2088-8708, DOI: 10.11591/ijece.v15i6.pp5986-6000 ❒ 5986

A systematic review of heuristic and meta-heuristic methods for dynamic task scheduling in fog computing environments

Hamed Talhouni[1,2], Noraida Haji Ali[1], Farizah Yunus[1], Saleh Atiewi[3], Yazrina Yahya[4]
[1]Faculty Computer Science and Mathematics, Universiti Malaysia Terengganu, Terengganu, Malaysia
[2]Department of Computer Information Systems, Al-Hussein Bin Talal University, Ma'an, Jordan
[3]Department of Computer Science, Al-Hussein Bin Talal University, Ma'an, Jordan
[4]Faculty Information Science and Technology, Universiti Kebangsaan Malaysia, Selangor, Malaysia

Article Info

Article history:

Received Aug 22, 2024
Revised Mar 27, 2025
Accepted May 24, 2025

Keywords:

Cloud computing
Fog computing
Heuristic
Meta heuristic
Task scheduling

ABSTRACT

The distributed fog node network and variable workloads make task distribution difficult in fog computing. Optimizing computing resources for dynamic workloads with heuristic and metaheuristic algorithms has shown potential. To address changing workloads, these algorithms enable real-time decision-making. This systematic review examines heuristic, meta-heuristic, and real-time dynamic job scheduling strategies in fog computing. Static methods like heuristic and meta-heuristic algorithms can help modify dynamic task scheduling in fog computing situations. This paper covers a current study area that stresses real-time approaches, meta-heuristics, and fog computing environments' dynamic nature. It also helps build reliable and scalable fog computing systems by spotting dynamic task scheduling trends, patterns, and issues. This study summarizes and analyzes the latest fog computing research on task-scheduling algorithms and their pros and cons to adequately address their issues. Fog computing task scheduling strategies are detailed and classified using a technical taxonomy. This work promises to improve system performance, resource utilization, and fog computing settings. The work also identifies fog computing job scheduling innovations and improvements. It reveals the strengths and weaknesses of present techniques, paving the way for fog computing research to address unresolved difficulties and anticipate future challenges.

Corresponding Author:

Noraida Haji Ali
Faculty Computer Science and Mathematics, Universiti Malaysia Terengganu
21030 Kuala Nerus, Terengganu Darul Iman, Malaysia
Email: aida@umt.edu.my

1. INTRODUCTION

Cloud computing offers varied nature of computing-oriented services such as software, networking, storage, and data analytics through internet connection to the internet of things (IoT) users. The application areas of IoT are augmented day by day, including smart agriculture, smart industry, military surveillance, and smart transportation. Hence, handling massive IoT tasks at the cloud layer causes huge latency and necessitates high network bandwidth which highly violates the service level agreement between the cloud service provider and IoT users as the distance between users and the cloud layer is quite far. To conquer these issues, fog computing is introduced as a boon to satisfy user requirements with low latency and energy. Fog computing (FC) emerges as a dynamic network paradigm aimed at addressing the challenges encountered by IoT users, particularly in terms of data processing speed and latency in retrieval. It is crucial to note that FC does not serve as a substitute for cloud computing (CC); rather, it complements and extends its capabilities.

Fog computing is situated near the IoT devices for storing and processing the data generated from those resource-constrained devices. As the CC is situated far away from the IoT devices, FC brings the computation and storage facilities much closer to the IoT devices. This proximity not only conserves energy but also ensures quality of service (QoS) for IoT devices. To improve QoS, fog computing systems have been widely integrated into IoT applications. The integration reduces reaction times to improve service delivery and operations [1]. Fog computing can be easily integrated into the IoT environment because of the growing popularity of virtualization in containers and the quick development of local computing devices [2]. However, when fog resources are scattered, resource-constrained, and various, effective task scheduling becomes essential for enhancing performance [3]. It is the cloud data center that is responsible for difficult duties because of its enormous processing and storage capacity, whereas activities that need rapid response especially those requiring crucial delays are delivered to the fog node due to its proximity [4].

Overall, fog computing refers to a decentralized configuration that combines a system of materials streams that is heavily automated with clients and cloud data centers sharing computing resources to improve their computational and data processing capabilities. Task-associated data analytics tackle several constraints, such as insufficient bandwidth and latency, and are presumed to be accomplished by completing data collection activities at the network edge [5]. Multiple methods have been used for optimizing bandwidth in task scheduling to reconcile conflicting demands, including quantity of data, processing specifications, security requirements, and network resource accessibility. Effectively combining task scheduling and bandwidth optimization is still a challenging problem despite recent achievements [6]. An example of highly dispersed processing is a cloud-fog framework, which consists of heterogeneous providers with a sense of a range of computational capabilities. For this reason, cloud-fog systems frequently utilize virtualization expertise to provide cloud and fog node resources in the form of virtual machines (VMs). It eliminates server heterogeneity, consolidates servers, and increases resource utilization rates [7]. However, the deployment of new virtualization technologies in fog computing task scheduling and resource allocation is affected by low latency services and scarce resources. In cloud computing, detailed consideration has been given to scheduling and load balancing. analyzed [8]. Allocation and scheduling have a direct impact on the system's quality of service, which includes energy usage and service time. Inefficiencies can lead performance to drop. Thus, selecting an effective fog computing node while preserving QoS is crucial [9]. Conversely, variation in energy consumption results from distinct task scheduling strategies on intelligent production lines where numerous fog nodes are powered by batteries; this invariably gives rise to a multitude of complications. For instance, research discovered that frequent data interchange, transmission, and processing might result in a considerable reduction in battery life through rapid consumption, hence creating a security risk for data leakage when a device is not charged in enough time [10]. The optimization problem of nondeterministic polynomial (NP) is the task scheduling method (nondeterministic polynomial time)-hard. To produce effective schedules in an acceptable period, many metaheuristics have been used, including the moth-flame optimization (MFO) algorithm, genetic algorithm (GA), and bees' life algorithm (BLA) [7]. NP-completeness characterizes the resource management algorithm, and its complexity varies with the time complexity. In fog computing, there are three ways to manage resources as efficiently as possible: heuristic, meta-heuristic, and hybrid methods. Large search spaces may be handled by meta-heuristic techniques, which can also find better resource management solutions in an acceptable length of time [11].

Optimization, a fundamental concept, permeates various aspects of everyday life in which the term optimization in computing refers to maximizing the network or performance of applications while using as few resources as possible. When it comes to optimization, population-level tactics modify a set of results as the procedure progresses, but meta-heuristic approaches are more effective at solving real-world issues in a range of fields, such as computer science and engineering [12]. Meta-heuristics are crucial to optimization. These algorithms are usually inspired by natural foraging and collective intelligence. The "African vulture optimization algorithm" (AVOA) is one of the novel meta-heuristics available. The AVOA is still in its early stages of development since it is a novel swarm intelligence technique. The optimization process is unpredictable, which makes it challenging for the AVOA to balance the phases of development and exploration. As a result, issues like a local optimum solution and a limited number of population states may arise for the method [13]. One "infrastructure as a service" (IaaS) cloud scheduling presentation is given. Following that attention is directed towards methods that are heuristic, meta-, and hyper-heuristic. Numerous problems that are computationally difficult may be effectively solved by using heuristic and meta-heuristic techniques. The main objective of the suggested technique is to show the benefits of heuristic methods, especially for problems involving task scheduling and resource allocation in the context of computers [14].

In this study, a systematic review is conducted by focusing on the heuristic and meta-heuristic methods employed in dynamic task scheduling within a fog computing environment. The study encompasses an in-depth analysis of task scheduling approaches, algorithms, and pivotal factors, drawing from a range of published studies spanning the period from 2021 to 2023. The primary objectives of this research are outlined as,

a. Providing an extensive overview of task scheduling methods within fog computing environments.
b. Review the technical aspects, such as architecture, benefits, and drawbacks, as well as the optimized methods and algorithms utilized in prior studies concerning dynamic task scheduling in fog computing.
c. In the context of the fog computing paradigm, we focus on heuristic and metaheuristic methodologies, with an examination of their applicability and usefulness in dynamic task scheduling procedures.
d. The goal is to identify and outline the current difficulties and challenges that are connected to task scheduling approaches in fog computing settings. This will be accomplished by identifying and defining the existing issues.

This paper is organized as follows. In Section 2, specifics of the task scheduling system used in the fog environment is discussed. Technical classification on task scheduling in a fog computing environment is defined in Section 3. Section 4 presents a theoretical examination and compression. The conclusion is finally presented in section 5.

2. TASK SCHEDULING ARCHITECTURE IN FOG COMPUTING

Fog computing was invented as a decentralized computing paradigm to address issues in the standard cloud computing systems such as high latency, bandwidth restrictions, and reduced energy efficiency by bringing networking, processing, and storage closer to the internet of things devices. Normally, the three layers of the fog computing architecture include the IoT/edge layer, the fog layer, and the cloud layer [15]. The internet of things layer is constituted by the sensors, actuators, and other internet of things endpoints that deliver data in real-time. Pre-processing and low-latency replies for resource-constrained devices are provided by the fog layer. The intermediary layer houses Fog nodes such as gateways, routers, and edge servers, thus relieving the cloud from the stress of processing and storing the data locally for time-sensitive operations. Fog nodes typically employ virtualization technologies like virtual machines or containers for providing scalability as well as effective consumption of resources. High-capacity data storage, sophisticated analytics, and worldwide coordination are handled by the cloud layer for jobs that do not need as much delay [16]. internet of things devices initiates the fog computing process by forwarding data to nearby fog nodes for on-site preprocessing and analysis. Work that requires large processing or long-term storage is shifted to the cloud layer. In this tiered setup, the hierarchical configuration brings in the benefits of bandwidth saving, latency reduction, and real-time decision-making, thereby bringing about QoS. This fog computing also saves energy as it avoids unnecessary data transfers to the cloud. It also allows for scalability due to the installation of more nodes in the fog as desired. Due to these characteristics, such features are very useful for latency-critical applications such as industrial automation, smart cities, and healthcare [11]. An explicitly cloud-fog-specific semi-dynamic real-time job scheduling framework. By efficiently allocating tasks, this algorithm reduces costs, makespan, and energy use. An adaptation of the grey wolf optimizer is shown to optimize task scheduling by taking into account several factors, including execution time, resource needs, and work duration [17]. resource aware prioritized task scheduling (RAPTS) is a task scheduling method used in a heterogeneous fog computing environment. The goal is to complete activities with deadline constraints on time, reduce reaction time and expense, as well as makespan, and maximize fog layer resource use [18].

Task scheduling is a widely acknowledged issue in fog computing environments, and different techniques exist to address it. Task scheduling algorithms include, for instance, genetic algorithms and other evolutionary techniques, as well as swarm intelligence-based strategies like ant colony optimization (ACO). These metaheuristic and approximated solutions draw inspiration from nature. Stated differently, they use non-deterministic and efficient methods to search the search space and identify the ideal parameters [19]. Overall, problems of task scheduling can be classified into two distinct classifications: dynamic scheduling and static scheduling. Static scheduling prevents the disclosure of all application task details earlier than their execution, while dynamic scheduling restricts access to information like this to runtime. Two types of algorithms are used for static scheduling: heuristic-based methods and supervised stochastic search-based techniques. The heuristic-based algorithms are within three categories: list-based, clustering, and job duplication-based [20]. The ideal location for the fog-to-cloud deployment of separate service components. To support user mobility and the dynamic QoS variables indicated earlier, separate processes must be carried out in an execution environment in this unstable [21].

Fog computing or cloud computing extended into edge devices, is thought to be crucial for effectively handling intelligent production line activities. Its high reliability, low latency, and distributed architecture allow it to react quickly to task requests from end devices. However, task requests in intelligent production lines typically need to be responded to quickly, therefore fog computing research must concentrate on dynamically scheduling activities to maximize fog nodes with limited resources. In intelligent manufacturing lines, this is crucial to lowering task reaction times and raising task completion rates [22].

A new strategy named HunterPlus investigates the impact of expanding the gated graph convolution networks (GGCNs) gated recurrent unit to a bidirectional gated recurrent unit. The article also examines the use of convolutional neural networks (CNNs) in cloud-fog optimization task scheduling [23]. A priority-based preemptive task scheduling with a multi-queue scheme that achieves an optimal task assignment for the delay-tolerant applications with a degree of processing delay and the latency-sensitive fog applications. At run-time, the multi-dimensional quantized polygon (MQP) algorithm divides tasks into short and long according to their burst time. The MQP algorithm keeps different task queues for different categories of tasks and adjusts the value of the time slot dynamically for preemption [24].

A parallel multi-threading platform is suggested to find the optimal offloading solution and the best sub-carrier for every offloaded task. Most importantly, our contribution binds a thread to every IoT device and creates a population of random solutions. Then, every population is updated and assessed based on the suggested fitness function that takes into account a tradeoff between delay and energy consumption. When new tasks are received in each time slot, an evaluation is conducted to preserve some members of the old population and to create new individuals based on some criteria [25]. A fog-cloud fit algorithm that distributes tasks between the fog and cloud in an even manner, depending on priority levels. Additionally, a modified Harris-hawks optimization (MHHO) inspired meta-heuristic method is suggested to allocate the optimal available resource to a task in a layer. The primary aim of this paper is to minimize the makespan time, task execution cost, and power consumption and maximize resource utilization in both the fog and cloud layers [26]. An algorithm that integrates a Hybrid task scheduling method in fog computing based on fuzzy logic and deep reinforcement learning (HTSFFDRL) algorithm with a Takagi-Sugeno fuzzy inference system. Through real-time interaction with the environment, this hybrid approach enables dynamic task prioritization as well as online modification of the scheduling rules [27].

An elephant herding optimization (EHO) algorithm based on Sine Cosine is integrated with the improved particle swarm optimization (IPSO) algorithm to improve the task scheduling performance using parameters such as load balancing and resource utilization. The traditional EHO and PSO algorithms are enhanced using a sine cosine-based clan-update operator and human group optimizer that enhance the algorithm's exploration and exploitation capabilities and prevent being caught in the local optima trap [28]. A hybrid evolutionary task scheduling and VM placement (HETSVP) algorithm for reliable fog computing task scheduling and VM placement. They solve the task execution time optimization and resource balance simultaneously by combining an enhanced particle swarm optimization algorithm with a novel VM placement strategy. They apply a binary encoding strategy, and the information about the position of the particle is represented using 0 and 1, while the particle velocity falls within the interval [0, 1]. When the scenario involves a discrete particle swarm, every particle's position will represent a potential plan for scheduling tasks. In addition to that, they supply the adaptive contraction factor that enhances the particle swarm optimization methodology [29].

A meta-heuristic algorithm hybrid grey wolf optimization (GWO) algorithm with particle swarm optimization (PSO) is called hybrid particle swarm optimization and grey wolf optimization (HPSO_GWO) to allocate the tasks to the VMs to maximize the QoS [30]. Examine security-conscious resource allocation in device-to-device-based fog computing systems. For improving task offloading, an innovative multi-objective function is introduced to optimize delay and energy savings over local computing as well as the cost of security breaches. Various multi-objective meta-heuristic algorithms, including non-dominated sorting genetic algorithm II (NSGA-II), have been introduced in the last decade. Maximizing objectives, like energy usage and delay, was the aim of utilizing these algorithms. Then the enhanced NSGA-II algorithm is utilized to find the solution to the problem. Sigma Scaling, a method to adjust the fitness values so that diversity is ensured in the population, is used to manage selection pressure in this algorithm. Incorporating Sigma Scaling, the algorithm's exploration and exploitation abilities are well managed to improve its potential to escape from local optima and avoid premature convergence [31].

A hybrid discrete optimization technique named HDSOS-GOA, utilizing the dynamic voltage and frequency scaling (DVFS) technique, is suggested to address scientific workflow scheduling issues in the fog computing paradigm. HDSOS-GOA integrates the search attributes of symbiotic organism's search (SOS) and grasshopper optimization algorithm (GOA) algorithms and the choice of these algorithms for executing workflow scheduling is determined by the probability computed by the learning automata. The HEFT technique is employed to calculate the task order [32]. A few categories for static and dynamic task-scheduling techniques are shown in Figure 1.

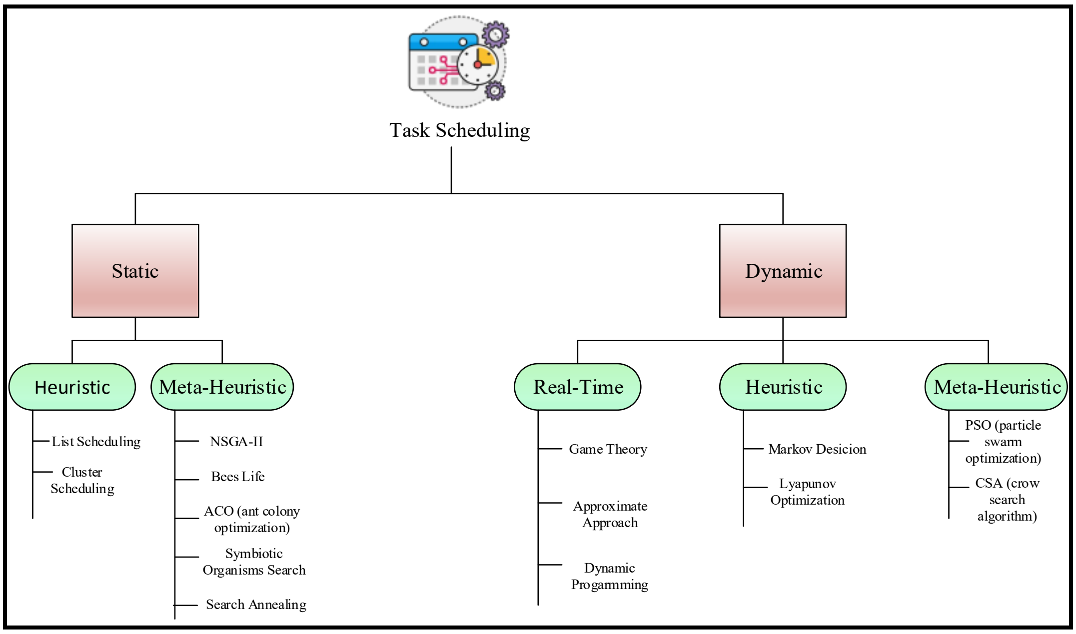

Figure 1. Classification of scheduling in fog computing

3. CLASSIFICATION OF DYNAMIC TASK SCHEDULING IN FOG COMPUTING

The following section explains a task-scheduling approach that operates within the context of fog computing. In addition, a comparative study will be carried out, which will include the examination of different techniques in a range of domains, such as the fundamental backdrop, case studies, advantages, disadvantages, and, finally, the specific results. A full research and rigorous analysis of heuristic and metaheuristic approaches for scheduling is being carried out, and this represents a component of that study. Figure 2 represents the fog computing architecture.

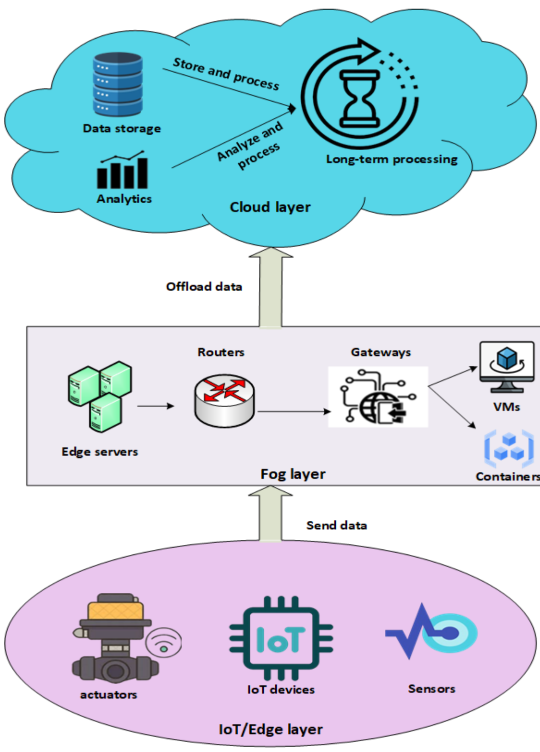

Figure 2. Architecture of fog computing

3.1. Dynamic task scheduling methods

Fog computing dynamic task scheduling is imperative because of the varying workload and limited resources. Rather than isolating real-time scheduling as an independent category, it can be considered a scheduling goal that is attainable using both heuristic and meta-heuristic techniques. Faster decision-making makes heuristic techniques best suited for real-time scheduling with less complex tasks. In contrast, meta-heuristic algorithms provide more optimization and flexibility to schedule efficiently in real-time for complicated and large-scale situations. Hence, the next section classifies scheduling techniques into heuristic-based and meta-heuristic-based real-time scheduling techniques, instead of real-time scheduling as a separate category.

3.1.1. Heuristic-based real-time scheduling

Heuristic-based scheduling algorithms apply rule-based or intuitive techniques to assign jobs in a fast manner. Such techniques are time-efficient for real-time applications as they execute in a very short time but often do not deliver optimal solutions. Priority-based scheduling, shortest-job-first, and round-robin scheduling are some examples.

3.1.2. Meta-heuristic-based real-time scheduling

Meta-heuristic techniques based on mathematical or natural-inspired processes provide wider exploration for ideal solutions. These methods like GA, PSO, and ACO are implemented to optimize the scheduling of tasks in real-time based on requirements such as task execution time, availability of resources, and power usage. This method is applied more in situations where there are a large size and dynamic environment of fog considering effective balancing between resource limitations and latency demands.

3.1.3. Heuristic techniques based on dynamic task scheduling

In a fog computing environment, heuristic-based scheduling techniques use intuitive or rule-of-thumb techniques to quickly and effectively decide how to divide up jobs among available resources. The heuristic scheduling strategy is the subject of two research. Hussain and Begh [33] proposed a new algorithm, hybrid flamingo search with a genetic algorithm (HFSGA), which lies in addressing the challenges associated with Transmitting information across a network, particularly the heightened Delay and the irregularity of the load. This hybrid heuristic algorithm aims to enhance the performance of task scheduling in fog computing by minimizing various costs, thereby improving quality of service (QoS). Among the key goals is the optimization of the creates as well as the process delay sensitivity time (PDST) in a fog cloud system. Using a new mix concerning Floating searching and genetic algorithms, HFSGA works towards the goal of providing cost-effective and quality service-aware task scheduling, which ultimately contributes to the enhancement of the resilience and stability of the larger process. The algorithm focuses on mitigating latency issues, managing burst traffic effectively, and contributing to the efficient allocation of tasks within the fog computing environment. Zhou [34] suggested a heuristic task scheduling approach for big data-driven fog computing environments that support machine learning. It builds a three-mode computing architecture and enhances a certain value taking reliability matrices, energy consumption, and latency into account. By improving crossover-mutation operators based on fitness values, the upgraded adaptive genetic algorithm improves on conventional techniques. The resulting fog resource scheduling in the dynamic landscape of intelligent manufacturing. Table 1 presents the classification of the preceding works along with their primary backgrounds, advancements, and drawbacks.

Table 2 describes how the above-mentioned publications were classified, including the used algorithm, QoS variables, and analytic environment. It offers a comprehensive view of the specific heuristic algorithms implemented in dynamic task scheduling scenarios. The table further illustrates how key QoS indicators such as latency, energy consumption, and execution time are addressed in various experimental setups. By categorizing these aspects, it becomes easier to identify performance patterns and potential optimization strategies for each algorithm. This classification enables researchers to systematically evaluate the suitability of different heuristic approaches under varied fog computing conditions.

Table 1. A comprehensive analysis of the advantages and limitations of the heuristic scheduling mechanism

Reference	Major context	Advantage	Limitation
[33]	Fog-cloud environment	– Superior QoS results. – Effective cost minimization. – Load balancing considerations.	Lack of consideration for security-related problems and load balancing issues in the fog-computing.
[35]	Intelligent manufacturing in the big data-driven	Comprehensive resource consideration. Enhance utilization.	Not investigate how fog computing's dynamic features affect its network and storage capacity.

Table 2. A comprehensive evaluation of the heuristic dynamic task scheduling method for QoS

Reference	Applied algorithm	QoS factors					Energy	Analysis environment
		Time	Cost	Latency	Scalability	Reliability		
[33]	Hybrid flamingo search with a genetic algorithm	✓	✓	✓	✓	×	×	Simulation (MATLAB)
[35]	Improved adaptive genetic algorithm	✓	×	×	×	✓	✓	Simulation (MATLAB)

3.1.4. Meta-heuristic method-based task scheduling approaches

The meta-heuristic method for task scheduling involves high-level strategies that guide the exploration of solution spaces to find optimal or near-optimal solutions. These methods are generic problem-solving frameworks that might be applied to many optimization issues, one of which is the scheduling of tasks in an environment using fog computing. We will analyze Ten studies in the meta-heuristic class. Kumar and Karri [36] proposed a nature-inspired multi-objective task scheduling algorithm called the electric earthworm optimization algorithm (EEOA) for IoT requests in a cloud-fog framework. This metaheuristic-based method offers a potential way to achieve service level agreement (SLA) demands while taking cloud-fog environments' complexity into account, thereby pushing the boundaries of task scheduling. Simulation results showed that the proposed method effectively balances the trade-offs between efficiency, cost, and energy consumption. To handle the complexity of the nonlinear issue, Najafizadeh *et al.* [37] proposed a multi-objective meta-heuristic method. The goal programming approach (GPA) emphasizes the requirement for safe, efficient, and economical job execution by employing limited solutions to meet multiple objectives. Moreover, the algorithm attains competitive outcomes concerning service cost, indicating its effectiveness in attaining a balanced and optimized resolution for job distribution in fog computing settings including internet of things devices. On the other hand, to optimize both the cost and the makespan, Apat *et al.* [38] suggested a traditional weighted multi-objective internet of things service allocation.

Khan *et al.* [15] described a cooperative method for managing fog nodes that makes use of the blockchain Hyperledger fabric and the B-drone genetic algorithm with metaheuristic support. Enhanced computing and processing efficiency are reached while lowering latency by using the proximity of unmanned aerial vehicles (UAVs) and fog nodes, making it feasible through a wireless sensor network. Comparing the B-drone approach's simulation results to other cutting-edge techniques, it is shown to enhance network robustness, decrease the cost of drone-ledger preservation, increase performance, and reduce computational costs. Basset *et al.* [39] a new model known as hybrid flamingo search with a genetic algorithm (HFSGA) is used in the proposed study to improve the scheduling of tasks. The effectiveness of HFSGA is compared with that of other well-known algorithms using seven fundamental benchmark optimization test functions. To further illustrate the importance of the findings, the Friedman rank test is used. Better results are shown by the adopted model in terms of makespan, cost, and percentage of deadline-satisfied tasks (PDST). Sing *et al.* [40] studied a whale optimization resource allocation (WORA), for a fog computing system prioritized for latency-sensitive IoT services. In a limited resource-based fog environment, WORA employs dynamic fuzzy c-mean clustering in the task classification and buffering module to handle heterogeneous real-time tasks, classifying and buffering using increased slack cost to divide concurrent computer threads. Moreover, the proposed algorithm was evaluated using evaluation metrics. The simulation outcomes demonstrated that this algorithm outperformed the competition in terms of average cost, and duration. Conversely, Shukla and Pandey [41] investigated the multi-objective artificial algae (MAA) algorithm, an effective meta-heuristic approach, to schedule scientific operations in a heterogeneous fog computing environment. To minimize execution times, energy consumption, and costs, tasks in the subsequent stage are scheduled using the MAA algorithm. Additionally, the algorithm employs an objective function based on weighted sums to optimize the utilization of fog resources.

Five benchmark scientific procedures are used to assess the proposed methodology. The suggested algorithm's performance is verified by contrasting its results with those of trained and traditional scheduling algorithms. The results show significant, trade-off-free improvements in execution time, energy usage, and overall cost over earlier approaches. Khaleel *et al.* [42] introduced an innovative algorithm that emphasizes multiple objectives in scheduling and computes a fitness function through ant colony optimization. To improve computational efficiency, the algorithm efficiently decides how to split up applications across edge and cloud servers. The results show that this strategy lowers energy usage and delays expenditures. Table 3 outlines the main background, and advantages. and drawbacks associated with metaheuristic-based dynamic scheduling techniques used in fog computing. Rahbari [43] several heuristic and meta-heuristic algorithms are examined in this study, and the hyper-heuristic scheduling (HHS) method is introduced to determine the optimal allocation in terms of low latency and energy usage. PEs are assigned to modules by HHS using low-level heuristics throughout the input workflow's training and testing stages.

Table 3. An in-depth analysis of the various meta-heuristic-based dynamic scheduling methodologies

Reference	Major context	Advantages	Limitation
[36]	Cloud-fog environment	– Improved QoS. – Effective handling of heterogeneous workload.	– Resource utilization is challenging. – Complexity.
[37]	Cloud-fog computing	– More economical with regard to the cost of the service, the oversight of the ability level, and the wait periods for the service.	– Limited privacy considerations. – It requires additional computational resources.
[38]	Fog computing environment	– Reduce makespan, cost, and energy.	– Not suitable for large-scale applications.
[15]	Fog environment	– Reduce computing cost. – Increase the performance. – Robust usage of the network.	– Limited bandwidth. – Increasing resource utilization. – Introduces complexities in long-term integrity.
[39]	Cloud Computing environment	– Improve both the exploration and exploitation. – Improve scaling factors.	– Algorithm implementation complexity.
[40]	Cloud-fog-based IoT applications	– Minimize the energy consumption, cost, and makespan.	– It does not explicitly consider the impact of network latency.
[41]	Heterogeneous fog-computing environment	– Improvement in execution time – Reduce energy consumption – Minimum cost	– Real-world development is challenging.
[42]	Edge-clod computing	– Minimizing delay. – Reducing energy usage.	– Complexity. – Real-world scenarios need further validation.
[44]	IoT in a collaborative cloud-fog environment	– Minimum cost, and makespan.	– Limited generalization. – Lack of dynamic world application.
[16]	IoT-fog-cloud network	– Minimization of makespan time.	– The cost of resource requesters is being disregarded.

Table 4 lists the categories for metaheuristic-based scheduling techniques, such as the applied algorithm, QoS considerations, and analytical environment. It highlights the diverse optimization methods utilized, including evolutionary and swarm intelligence approaches tailored for fog computing. Each listed algorithm is evaluated in terms of its ability to address critical QoS metrics like time, cost, scalability, and reliability under simulation environments. This detailed categorization aids in comparing the practical efficiency of different metaheuristic strategies across a broad spectrum of fog computing contexts. Such insights are crucial for selecting the most appropriate algorithm in complex, dynamic task scheduling scenarios where real-time performance and energy efficiency are paramount.

Table 4. An in-depth analysis of the meta-heuristic dynamic scheduling technique for QoS

Reference	Applied algorithm	QoS Factors					Energy	Analysis environment
		Time	Cost	Latency	Scalability	Reliability		
[36]	EEOA	✓	✓	×	✓	×	✓	Simulation (Improved cuckoo and OBL)
[37]	Multi-objective simulated annealing	✓	✓	×	×	×	×	Simulation (MATLAB)
[38]	FSPGA, FSPSA, and FSPSO	✓	✓	✓	×	×	✓	Simulation (Yet Another Fog Simulator (YAFS))
[15]	Meta-heuristic-enabled genetic algorithm	✓	✓	×	×	×	✓	Simulation
[39]	Hybrid differential evolution (HDE)	✓	✓	×	×	✓	×	Simulation (CloudSim platform)
[40]	WORA	✓	✓	✓	×	×	✓	Simulation (Python)
[41]	MAA	✓	✓	×	✓	✓	✓	Simulation (Java IDE Eclipse)
[42]	ACO	✓	×	✓	×	×	✓	Simulation (Java-based language)
[44]	Whale optimization algorithm (WOA)	×	✓	×	✓	✓	✓	Simulation (VM workstation)
[16]	Hybrid meta-heuristic (MH) algorithm	✓	×	×	×	×	×	Simulation (MATLAB R2018b software)

Kumar *et al.* [44] have suggested an artificial intelligence (AI) framework that integrates fuzzy models to make intelligent decisions about the choice of working distantly on data centers in the cloud or fog nodes or nearby on-edge devices to complete tasks. In the collaborative cloud-fog scenario, the solution uses the WOA, a metaheuristic technique, to achieve optimum task-to-resource mapping. Extensive experimental analysis establishes the effectiveness about the effectiveness of the suggested strategy in enhancing the general efficacy of industry 5.0 deadline-aware assignments, with substantial gains being noticed in makespan, execution cost, rejection ratio, and energy usage when compared with alternative approaches. Liu *et al.* [16] describe AO_AVOA, a hybrid meta-heuristic (MH) method that uses Aquila optimizer (AO) and the African vultures optimization method (AVOA) to schedule IoT requests in IoT fog-clod networks. By using AO operators to locate the optimum solution while attempting to identify the ideal scheduling solution, AO_AVOA enhances the AVOA explore phase. Based on performance metrics consisting of makespan and throughput, AO_AVOA proved to be highly effective in solving the scheduling problem in IoT-fog-cloud networks when compared to methods of AVOA, AO, firefly algorithm (FA), particle swarm optimization (PSO), and Harris hawks optimization (HHO).

3.1.5. Real-time-based dynamic task scheduling

Four different research related to real-time scheduling were looked at. A fuzzy logic-based task scheduling technique that divides up the workload in a fog-cloud computing system between the fog and cloud layers was presented by Ali *et al.* [45]. The method selects the appropriate process to do the given job by using the tsk. The simulation's results show that the suggested strategy lowers the average turnaround time, makespan time, and delay rate. Real time heterogeneous hierarchical scheduling (RTH2S), a scheduling algorithm proposed by Heuvel [46], is designed to manage a collection of real-time tasks within a heterogeneous integrated fog-cloud architecture. The algorithm divides the task according to its size and deadline requirements, or it selects a preferable fog node for the task's execution. The suggested method's simulation results show lower costs and higher success rates. In another work, Mattia and Beraldi [47] presented a decentralized approach for real-time dynamic scheduling in fog computing environments that is based on reinforcement learning (RL). Each fog node was designed to have this algorithm installed to enable autonomous scheduling choices depending on the circumstances at hand. The classification of the aforementioned studies, including their primary context, advantages, and limitations, is explained in Table 5.

The outcomes of the delay-based simulation demonstrated that, under both fixed load conditions and real-world geographic scenarios, there is a minimum cost to complete within the deadline in a work that is equal to every node. Ranjan and Sharma [48] suggested an approach that presents two schedulers based on nonlinear mathematical programming: one for cloud computing and another for fog computing. These schedulers challenge pre-established conventions by allocating tasks depending on attributes like performance and availability. Fuzzy logic is used in this strategy to divide work across the cloud and fog layers based on task needs (computing, storage, bandwidth), as well as constraints (size of data, deadlines). The strategy outperforms current methods in terms of completed tasks, average turnaround time, calculation time, and latency rate, as shown by simulation testing. Table 6 delineates the criteria used to classify the aforementioned articles, including the analytic environment, QoS elements, and applied method.

Table 5. A comprehensive comparison of real-time task scheduling algorithms

Reference	Major context	Advantage	Limitation
[45]	Fog-cloud computing	– It reduces the waiting time of the tasks. – The load balancing mechanism enhances the overall efficiency	– Need for further investigation on channel bandwidth. – Incomplete consideration of privacy and mobility
[46]	Hierarchical heterogeneous fog networks	Efficient resource utilization. Minimize the overall cost	Variations in workload patterns can be challenging
[47]	Smart cities	Maximize the task execution time.	Increased complexity of system state
[48]	Fog computing using deep learning techniques	Increase the efficiency of execution of tasks in IoT systems	Low scalability

Table 6. An in-depth analysis of the real-time scheduling technique for QoS

Reference	Applied algorithm	QoS factors					Energy	Analysis environment
		Time	Cost	Latency	Scalability	Reliability		
[45]	Fuzzy logic-based task scheduling algorithm	✓	×	✓	×	×	×	Simulation (iFogSim)
[46]	Real-time heterogeneous hierarchical scheduling	✓	✓	×	×	×	×	Simulation (iFogSim)
[47]	Reinforcement learning	×	✓	×	×	×	×	Simulation (delay-based simulator)
[48]	Fuzzy logic	✓	✓	×	×	×	×	Simulation (iFogSim)

Table 7 represents the comparison between the heuristic and meta-heuristic algorithms. Heuristic methods are quicker and less complex and are therefore preferable for real-time, low-complexity fog computing scheduling. Meta-heuristic algorithms are more adaptive and powerful and therefore can be implemented for optimizing huge scheduling and dynamic resource management. Therefore, for real-time scheduling of tasks, heuristic solutions are effective whereas for highly constrained scheduling and optimization, meta-heuristic algorithms have better outcomes while being computationally expensive.

Table 7. Comparison of heuristic and meta-heuristic algorithms in fog computing

Feature	Heuristics algorithm	Meta-heuristic algorithm
Definition	Problem-specific rules or logic-based approaches that provide a quick solution but may not be optimal	General-purpose optimization techniques inspired by nature or mathematics to explore better solutions
Search mechanism	Uses a predefined set of rules to find solutions	Uses stochastic or guided random searches to find optimal or near-optimal solutions
Optimization level	Works well for small-scale problems but struggles with complex, large-scale tasks.	Suitable for large-scale, complex problems as it explores a broader solution space
Adaptability	Less flexible as it follows a fixed approach to problem-solving	More flexible and adaptive as it dynamically adjusts search strategies
Computational complexity	Low complexity; executes faster but may get stuck in local optima	Higher complexity; requires more processing power but avoids local optima
Solution quality	Provides an approximate solution that is fast but may not be optimal	Provides near-optimal or optimal solutions with better accuracy
Examples	- First come first serve (FCFS) - Shortest job first (SJF) - Round Robin (RR) - Priority scheduling	- Genetic algorithm (GA) - Particle swarm optimization (PSO) - Ant colony optimization (ACO) - Grey wolf optimizer (GWO)
Usage in fog computing	Suitable for simpler, real-time task scheduling scenarios.	Preferred for complex scheduling and resource allocation problems to balance latency, energy, and cost
Main drawback	Can lead to suboptimal solutions due to its greedy approach	Requires more computation time and tuning to achieve the best results

4. ANALYTICAL DISCUSSION

This section provides an analytic account of the problems with task scheduling in a fog computing environment. It addresses issues about several dimensions such as resource constraints and dynamic scenarios that impact the success in applying scheduling strategies. Specific concerns are also made around how to adapt scheduling technologies so that they fit and match the specific requirements, needs, and conditions faced in a fog computing system demand for scalability, energy efficiency, and integration with edge AI advancements.

4.1. Analytical discussion

This section provides an in-depth examination of the categories of job scheduling algorithms, thereby providing knowledge about the several techniques present in fog computing. An assessment is performed in light of task scheduling research by which better knowledge can be obtained about the way through which these algorithms handle a particular difficulty presented by fog scenarios. Analysis charts are also employed to represent the quality-of-service factors by depicting the influential and essential performance indicators and how these will impact the effectiveness of task scheduling in situations regarding fog computing.

Figure 3 dynamic scheduling methods categories in fog environment shows how job scheduling strategies are categorized in a fog computing environment. The most popular job scheduling techniques are metaheuristic-based algorithms, which have a 62.5% utilization rate. More than any other algorithm, heuristic methods have been implemented in fog computing to obtain the most effective approach for performing tasks or jobs. It is important to highlight that when it comes to fog computing, formal techniques can be a valuable tool for validating the correctness of dynamic scheduling approaches and evaluating functional properties.

The well-known simulation programs CloudSim and iFogSim control the computational environment in fog computing studies. These tools offer a thorough platform for researchers and include all approaches used in task scheduling investigations. These simulation settings are the main place where scheduling algorithms are tested and verified because of the difficulties and complexities associated with real-world implementation.

The QoS elements in fog environment experiments are shown in Figure 4 QoS considerations in fog computing task scheduling research. The most significant component, with a 36% utilization rate, is time. Certain difficult tasks, such as evaluating privacy issues, scalability, and reliability that have not been assessed in fog computing, are involved in assessing the effectiveness of scheduling.

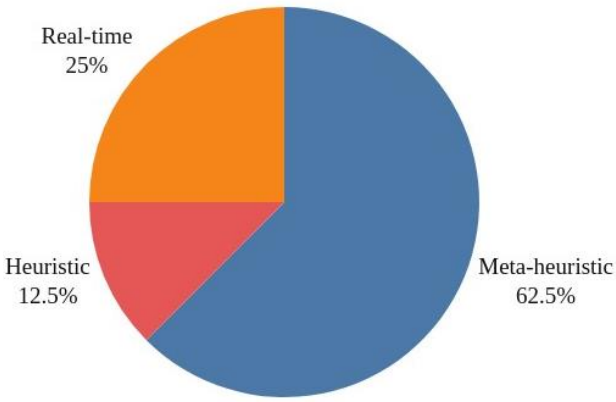

Figure 3. Dynamic scheduling methods categories in fog environment

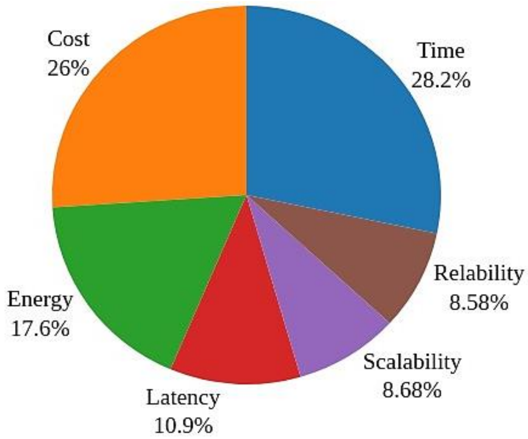

Figure 4. QoS considerations in fog computing task scheduling research

4.2. Challenges

In the following section, some of the main challenges that prevent implementing task scheduling in fog computing are discussed. One such challenges that come with these difficulties is flexibility in dynamic settings since variations in workload patterns, network circumstances, and availability of devices might impact the performance. Moreover, with increasing numbers of devices and jobs, scalability and efficiency will become critical factors that would demand algorithms that can control huge deployments without sacrificing too much efficiency. Further difficulties in designing effective task scheduling solutions for fog computing are brought into sharp relief by considering problems such as energy efficiency, security requirements, and real-time bounds.

a. Adaptability to dynamic environments: Many fog computing environments are dynamic, as seen by changes in device availability, network conditions, and workload patterns. Ensuring the flexibility of heuristic and meta-heuristic approaches to accommodate changing conditions is a critical concern.
b. Efficiency and scalability: In fog computing networks, the scalability of heuristic and metaheuristic approaches becomes critical as the number of devices and tasks increases. It is necessary to have effective algorithms that can manage extensive deployments while preserving acceptable efficiency.
c. Limitations on real-time: Fog computing applications are typical examples of those that need a high level of real-time. It is a constantly growing field of study how effectively heuristic and meta-heuristic techniques can manage these real-time constraints, especially in essential applications like industrial automation or healthcare.
d. Major security considerations: Scheduling methods based on heuristics and meta-heuristics may be subject to security flaws that need careful investigation. The reliability of fog computing systems must not be jeopardized by security lapses or other risks.
e. Energy effectiveness: Energy conservation must be given priority in heuristic and metaheuristic techniques because of the restricted resources present in many fog devices. The degree to which algorithms can be created to strike a balance between performance and energy efficiency is still up for debate.
f. Collaboration with edge AI: The increasing usage of edge AI in fog computing necessitates greater study into the seamless integration of heuristic and metaheuristic scheduling approaches with edge AI algorithms to enhance edge decision-making.

5. CONCLUSION

This study comprehensively analyzed all recent publications from the years 2021 to 2023 concerning task scheduling methods in fog computing. It delved into specific techniques, scrutinized their algorithms, and evaluated their outcomes in simulation applications. The analysis encompassed factors such as task scheduling categories and various algorithms utilized for scheduling tasks.

Within the parameters of the investigation, task scheduling techniques were separated into two groups: static and dynamic approaches. The use of real-time, meta-heuristic, and heuristic algorithms in the dynamic domain was examined. Consequently, meta-heuristic algorithms are the most widely applied algorithms for job scheduling, with a consumption rate of 65.5%. The execution time element is the most essential contributing to 28.3% of the total. We have provided information on heuristic and meta-heuristic approaches for dynamic task scheduling in fog computing environments through our thorough investigation and systematic evaluation. Future research in this area should examine how machine-learning approaches might be integrated for better performance and adaptability. Energy-efficient scheduling that focuses on consumption reduction is still a crucial area of research. Furthermore, hybrid approaches that combine various optimization techniques with heuristic/metaheuristic methods and the creation of adaptable, self-organizing systems might have synergistic benefits. Enhancing the practical usability and effect of dynamic work scheduling solutions in fog computing environments.

ACKNOWLEDGMENTS

The authors would like to express their sincere gratitude to Universiti Malaysia Terengganu (UMT) for their support and encouragement throughout the course of this research. The resources, academic environment, and institutional backing provided by UMT have been instrumental in the successful completion of this work

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Hamed Talhouni	✓	✓		✓	✓			✓	✓	✓			✓	
Noraida Haji Ali	✓	✓	✓			✓		✓	✓	✓		✓		✓
Farizah Yunus			✓		✓	✓			✓					
Saleh Atiewi							✓		✓		✓	✓		
Yazrina Yahya					✓	✓	✓			✓		✓		

C : **C**onceptualization	I : **I**nvestigation	Vi : **Vi**sualization
M : **M**ethodology	R : **R**esources	Su : **Su**pervision
So : **So**ftware	D : **D**ata Curation	P : **P**roject administration
Va : **Va**lidation	O : Writing - **O**riginal Draft	Fu : **Fu**nding acquisition
Fo : **Fo**rmal analysis	E : Writing - Review & **E**diting	

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, NH, upon reasonable request.

REFERENCES

[1] H. Tran-Dang and D.-S. Kim, "Dynamic collaborative task offloading for delay minimization in the heterogeneous fog computing systems," *Journal of Communications and Networks*, vol. 25, no. 2, pp. 244–252, Apr. 2023, doi: 10.23919/JCN.2023.000008.
[2] P. Krivic, M. Kusek, I. Cavrak, and P. Skocir, "Dynamic scheduling of contextually categorised internet of things services in fog computing environment," *Sensors*, vol. 22, no. 2, p. 465, Jan. 2022, doi: 10.3390/s22020465.
[3] J. U. Arshed, M. Ahmed, T. Muhammad, M. Afzal, M. Arif, and B. Bazezew, "GA-IRACE: genetic algorithm-based improved resource aware cost-efficient scheduler for cloud fog computing environment," *Wireless Communications and Mobile Computing*, vol. 2022, no. 1, Jan. 2022, doi: 10.1155/2022/6355192.
[4] F. A. Saif, R. Latip, Z. M. Hanapi, and K. Shafinah, "Multi-objective grey wolf optimizer algorithm for task scheduling in cloud-fog computing," *IEEE Access*, vol. 11, pp. 20635–20646, 2023, doi: 10.1109/ACCESS.2023.3241240.
[5] G. Shruthi, M. R. Mundada, B. J. Sowmya, and S. Supreeth, "Mayfly Taylor optimisation-based scheduling algorithm with deep reinforcement learning for dynamic scheduling in fog-cloud computing," *Applied Computational Intelligence and Soft Computing*, vol. 2022, pp. 1–17, Aug. 2022, doi: 10.1155/2022/2131699.
[6] Z. Yin *et al.*, "A multi-objective task scheduling strategy for intelligent production line based on cloud-fog computing," *Sensors*,

vol. 22, no. 4, p. 1555, Feb. 2022, doi: 10.3390/s22041555.
[7] A. S. Abohamama, A. El-Ghamry, and E. Hamouda, “Real-time task scheduling algorithm for IoT-based applications in the cloud–fog environment,” *Journal of Network and Systems Management*, vol. 30, no. 4, p. 54, Oct. 2022, doi: 10.1007/s10922-022-09664-6.
[8] F. M. Talaat, H. A. Ali, M. S. Saraya, and A. I. Saleh, “Effective scheduling algorithm for load balancing in fog environment using CNN and MPSO,” *Knowledge and Information Systems*, vol. 64, no. 3, pp. 773–797, Mar. 2022, doi: 10.1007/s10115-021-01649-2.
[9] S. S. Hajam and S. A. Sofi, “Spider monkey optimization based resource allocation and scheduling in fog computing environment,” *High-Confidence Computing*, vol. 3, no. 3, p. 100149, Sep. 2023, doi: 10.1016/j.hcc.2023.100149.
[10] T. Ahanger, F. Dahan, U. Tariq, and I. Ullah, “Quantum inspired task optimization for IoT edge fog computing environment,” *Mathematics*, vol. 11, no. 1, p. 156, Dec. 2022, doi: 10.3390/math11010156.
[11] M. A. Ibrahim and S. Askar, “An intelligent scheduling strategy in fog computing system based on multi-objective deep reinforcement learning algorithm,” *IEEE Access*, vol. 11, pp. 133607–133622, 2023, doi: 10.1109/ACCESS.2023.3337034.
[12] B. V Natesha and R. M. R. Guddeti, “Meta-heuristic based hybrid service placement strategies for two-level fog computing architecture,” *Journal of Network and Systems Management*, vol. 30, no. 3, p. 47, Jul. 2022, doi: 10.1007/s10922-022-09660-w.
[13] R. Ghafari and N. Mansouri, “E-AVOA-TS: enhanced African vultures optimization algorithm-based task scheduling strategy for fog–cloud computing,” *Sustainable Computing: Informatics and Systems*, vol. 40, p. 100918, Dec. 2023, doi: 10.1016/j.suscom.2023.100918.
[14] R. Aron and A. Abraham, “Resource scheduling methods for cloud computing environment: the role of meta-heuristics and artificial intelligence,” *Engineering Applications of Artificial Intelligence*, vol. 116, p. 105345, Nov. 2022, doi: 10.1016/j.engappai.2022.105345.
[15] A. A. Khan *et al.*, “A drone-based data management and optimization using metaheuristic algorithms and blockchain smart contracts in a secure fog environment,” *Computers and Electrical Engineering*, vol. 102, p. 108234, Sep. 2022, doi: 10.1016/j.compeleceng.2022.108234.
[16] Q. Liu, H. Kosarirad, S. Meisami, K. A. Alnowibet, and A. N. Hoshyar, “An optimal scheduling method in IoT-fog-cloud network using combination of aquila optimizer and African vultures optimization,” *Processes*, vol. 11, no. 4, p. 1162, Apr. 2023, doi: 10.3390/pr11041162.
[17] A. Satouf, A. Hamidoğlu, Ö. M. Gül, A. Kuusik, L. Durak Ata, and S. Kadry, “Metaheuristic-based task scheduling for latency-sensitive IoT applications in edge computing,” *Cluster Computing*, vol. 28, no. 2, p. 143, Apr. 2025, doi: 10.1007/s10586-024-04878-6.
[18] M. Hussain, S. Nabi, and M. Hussain, “RAPTS: resource aware prioritized task scheduling technique in heterogeneous fog computing environment,” *Cluster Computing*, vol. 27, no. 9, pp. 13353–13377, Dec. 2024, doi: 10.1007/s10586-024-04612-2.
[19] J. Lim, “Latency-aware task scheduling for IoT applications based on artificial intelligence with partitioning in small-scale fog computing environments,” *Sensors*, vol. 22, no. 19, p. 7326, Sep. 2022, doi: 10.3390/s22197326.
[20] R. Madhura, B. L. Elizabeth, and V. R. Uthariaraj, “An improved list-based task scheduling algorithm for fog computing environment,” *Computing*, vol. 103, no. 7, pp. 1353–1389, Jul. 2021, doi: 10.1007/s00607-021-00935-9.
[21] P. Maiti, B. Sahoo, A. K. Turuk, A. Kumar, and B. J. Choi, “Internet of Things applications placement to minimize latency in multi-tier fog computing framework,” *ICT Express*, vol. 8, no. 2, pp. 166–173, Jun. 2022, doi: 10.1016/j.icte.2021.06.004.
[22] F. Xu, Z. Yin, Y. Li, F. Zhang, and G. Xu, “The task scheduling algorithm for fog computing in intelligent production lines based on DQN,” in *2023 15th International Conference on Communication Software and Networks (ICCSN)*, Jul. 2023, pp. 449–455, doi: 10.1109/ICCSN57992.2023.10297319.
[23] S. Iftikhar *et al.*, “HunterPlus: AI based energy-efficient task scheduling for cloud–fog computing environments,” *Internet of Things*, vol. 21, p. 100667, Apr. 2023, doi: 10.1016/j.iot.2022.100667.
[24] M. Fahad, M. Shojafar, M. Abbas, I. Ahmed, and H. Ijaz, “A multi-queue priority-based task scheduling algorithm in fog computing environment,” *Concurrency and Computation: Practice and Experience*, vol. 34, no. 28, Dec. 2022, doi: 10.1002/cpe.7376.
[25] S. I. AlShathri, S. A. Chelloug, and D. S. M. Hassan, “Parallel meta-heuristics for solving dynamic offloading in fog computing,” *Mathematics*, vol. 10, no. 8, p. 1258, Apr. 2022, doi: 10.3390/math10081258.
[26] I. Z. Yakubu and M. Murali, “An efficient meta-heuristic resource allocation with load balancing in IoT-Fog-cloud computing environment,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 3, pp. 2981–2992, Mar. 2023, doi: 10.1007/s12652-023-04544-6.
[27] P. Choppara and S. S. Mangalampalli, “A hybrid task scheduling technique in fog computing using fuzzy logic and deep reinforcement learning,” *IEEE Access*, vol. 12, pp. 176363–176388, 2024, doi: 10.1109/ACCESS.2024.3505546.
[28] K. J. Rajashekar, Channakrishnaraju, P. C. Gowda, and A. B. Jayachandra, “SCEHO-IPSO: a nature-inspired meta heuristic optimization for task-scheduling policy in cloud computing,” *Applied Sciences*, vol. 13, no. 19, p. 10850, Sep. 2023, doi: 10.3390/app131910850.
[29] D. Yu and W. Zheng, “A hybrid evolutionary algorithm to improve task scheduling and load balancing in fog computing,” *Cluster Computing*, vol. 28, no. 1, p. 74, Feb. 2025, doi: 10.1007/s10586-024-04749-0.
[30] K. C. Dev, B. B. Dash, U. C. De, P. Pattnayak, R. Satapthy, and S. S. Patra, “Task scheduling in fog assisted cloud environment using hybrid metaheuristic algorithm,” in *International Conference on Expert Clouds and Applications*, 2023, pp. 223–236, doi: 10.1007/978-981-99-1745-7_16.
[31] M. Mohammadi, F. BahraniPour, S. Ebrahimi Mood, and M. Farshi, “Security-aware resource allocation in fog computing using a meta-heuristic algorithm,” *Cluster Computing*, vol. 28, no. 2, p. 104, Apr. 2025, doi: 10.1007/s10586-024-04794-9.
[32] A. Mohammadzadeh, M. Akbari Zarkesh, P. Haji Shahmohamd, J. Akhavan, and A. Chhabra, “Energy-aware workflow scheduling in fog computing using a hybrid chaotic algorithm,” *The Journal of Supercomputing*, vol. 79, no. 16, pp. 18569–18604, Nov. 2023, doi: 10.1007/s11227-023-05330-z.
[33] S. M. Hussain and G. R. Begh, “Hybrid heuristic algorithm for cost-efficient QoS aware task scheduling in fog–cloud environment,” *Journal of Computational Science*, vol. 64, p. 101828, Oct. 2022, doi: 10.1016/j.jocs.2022.101828.
[34] R. Zhou, “A heuristic task scheduling strategy for intelligent manufacturing in the big data-driven fog computing environment,” *Mobile Information Systems*, vol. 2022, pp. 1–10, Aug. 2022, doi: 10.1155/2022/5830760.
[35] R. Zhou, “(Double) A heuristic task scheduling strategy for intelligent manufacturing in the big data-driven fog computing environment,” *Mobile Information Systems*, vol. 2022, pp. 1–10, Aug. 2022, doi: 10.1155/2022/5830760.
[36] M. S. Kumar and G. R. Karri, “EEOA: cost and energy efficient task scheduling in a cloud-fog framework,” *Sensors*, vol. 23, no. 5, p. 2445, Feb. 2023, doi: 10.3390/s23052445.

[37] A. Najafizadeh, A. Salajegheh, A. M. Rahmani, and A. Sahafi, “Multi-objective task scheduling in cloud-fog computing using goal programming approach,” *Cluster Computing*, vol. 25, no. 1, pp. 141–165, Feb. 2022, doi: 10.1007/s10586-021-03371-8.
[38] H. K. Apat, B. Sahoo, V. Goswami, and R. K. Barik, “A hybrid meta-heuristic algorithm for multi-objective IoT service placement in fog computing environments,” *Decision Analytics Journal*, vol. 10, p. 100379, Mar. 2024, doi: 10.1016/j.dajour.2023.100379.
[39] M. Abdel-Basset, R. Mohamed, W. Abd Elkhalik, M. Sharawi, and K. M. Sallam, “Task scheduling approach in cloud computing environment using hybrid differential evolution,” *Mathematics*, vol. 10, no. 21, p. 4049, Oct. 2022, doi: 10.3390/math10214049.
[40] R. Sing, S. K. Bhoi, N. Panigrahi, K. S. Sahoo, N. Jhanjhi, and M. A. AlZain, “A whale optimization algorithm based resource allocation scheme for cloud-fog based IoT applications,” *Electronics*, vol. 11, no. 19, p. 3207, Oct. 2022, doi: 10.3390/electronics11193207.
[41] P. Shukla and S. Pandey, “MAA: multi-objective artificial algae algorithm for workflow scheduling in heterogeneous fog-cloud environment,” *The Journal of Supercomputing*, vol. 79, no. 10, pp. 11218–11260, Jul. 2023, doi: 10.1007/s11227-023-05110-9.
[42] M. I. Khaleel, M. Safran, S. Alfarhood, and D. Gupta, “Combinatorial metaheuristic methods to optimize the scheduling of scientific workflows in green DVFS-enabled edge-cloud computing,” *Alexandria Engineering Journal*, vol. 86, pp. 458–470, Jan. 2024, doi: 10.1016/j.aej.2023.11.074.
[43] D. Rahbari, “Analyzing meta-heuristic algorithms for task scheduling in a fog-based IoT application,” *Algorithms*, vol. 15, no. 11, p. 397, Oct. 2022, doi: 10.3390/a15110397.
[44] M. Kumar, G. K. Walia, H. Shingare, S. Singh, and S. S. Gill, “AI-based sustainable and intelligent offloading framework for IIoT in collaborative cloud-fog environments,” *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 1414–1422, Feb. 2024, doi: 10.1109/TCE.2023.3320673.
[45] H. S. Ali, R. R. Rout, P. Parimi, and S. K. Das, “Real-time task scheduling in fog-cloud computing framework for IoT applications: a fuzzy logic based approach,” in *2021 International Conference on COMmunication Systems & NETworkS (COMSNETS)*, Jan. 2021, pp. 556–564, doi: 10.1109/COMSNETS51098.2021.9352931.
[46] V. Den Heuvel, “ORCA online research @ cardiff.” pp. 1–2, 2023.
[47] G. P. Mattia and R. Beraldi, “On real-time scheduling in fog computing: a reinforcement learning algorithm with application to smart cities,” in *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, Mar. 2022, pp. 187–193, doi: 10.1109/PerComWorkshops53856.2022.9767498.
[48] V. Ranjan and L. Sharma, “Real-time task scheduling and resource scheduling in fog computing using deep learning techniques,” in *2023 International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, Apr. 2023, pp. 1–6, doi: 10.1109/ICDCECE57866.2023.10150474.

BIOGRAPHIES OF AUTHORS

Hamed Talhouni has been employed as a lecturer at Al Hussein bin Tala University (AHU) since 2014. He is presently the Dean's assistant for student affairs. He earned his B.Sc. degree in computer science in 2004 and his M.Sc. degree in computer information systems in 2008. He is presently pursuing a Ph.D. at Universiti Malaysia Terengganu (UMT). His research interests encompass information retrieval, cloud computing, fog computing, and the internet of things (IoT). He can be contacted at email: p4650@pps.umt.edu.my.

Noraida Haji Ali received her bachelor's, master's degree, and Ph.D. in computer science from Universiti Kebangsaan Malaysia. She has been a lecturer in the Faculty of Computer Science and Mathematics at Universiti Malaysia Terengganu since 2000. Her current research interests focus on the software engineering area, especially systems development, decision-support systems, and object-oriented modelling, including formal methods and e-learning. She has presented and published many papers in the software engineering area at various international and local refereed journals, symposiums, and conferences. She is actively conducting her research in her field of interest under the supervision of undergraduate and postgraduate students. She has already completed a few projects under research grants from the Ministry of Higher Education (MOHE) as a leader and co-researcher. Now, she actively does the few research projects that she collaborates on with a few industries. She is also actively involved in consulting work for outside organizations. She was invited as a member to attend a focus group discussion about academic assessment, including as a member of the journal publisher, program curriculum evaluator, instructor for the communities, and entrepreneurship. She is actively involved in various committees. Internationally, she has also been invited to be one of the organizing committees for several international conferences. Her contribution to scientific computing has been recognized by receiving several awards, including the product and research innovation award for international and national levels, the UMT excellence service award, the excellent teaching award, and the article publication award. She is a member of the MBOT professional technologists, Malaysian software engineering interest group (MySIG) and mobile learning association of Malaysia. She can be contacted at email: aida@umt.edu.my.

Farizah Yunus received her B.Sc. degree in electrical engineering (telecommunication) and PhD degrees in telecommunication engineering from Universiti Teknologi Malaysia (UTM). She is a senior lecturer of computer science at faculty of computer science and mathematics, Universiti Malaysia Terengganu (UMT). Her research interests include wireless sensor network, cloud computing, internet of things (IoT) and cyber security. She is a member of MBOT and BEM. She has worked as researcher in several national funded R&D projects. She can be contacted at email: farizah.yunus@umt.edu.my.

Saleh Atiewi received his B.Sc. degree in computer science from Al-Isra University, Amman, Jordan, in 1999, followed by a master's degree in internet technology from Wollongong University, Wollongong, Australia, in 2004. He later earned his Ph.D. in computer science from Tenaga Nasional University, Putrajaya, Malaysia, in 2017. Since 2004, Dr. Atiewi has been part of Al Hussein Bin Talal University in Ma'an, Jordan. He is currently an associate professor in the Department of Computer Science, actively contributing to the university's academic and research initiatives. Over the years, Dr. Atiewi has held several key administrative and leadership roles, including: head of the computer science department, vice-dean of scientific research and postgraduate studies, director of the computer center and information technology, and director of the center for innovation, creativity, and entrepreneurship Dr. Atiewi's research interests include network security, cloud computing, security, and the internet of things (IoT). He remains deeply committed to advancing knowledge and fostering innovation within his field. He can be contacted at email: saleh@ahu.edu.jo.

Yazrina Yahya is currently an associate professor at the faculty of information science and technology at Universiti Kebangsaan Malaysia (UKM) where her academic career began in 1997. She was recently seconded to the Higher Education Leadership Academy (AKEPT) as the deputy director for center leadership sustainability. Her task at AKEPT requires her to establish global leadership management programs that will assist the leaders and communities of local and ASEAN academic institutions She has been appointed as the governing board member of SEAMEO RIHED, responsible for operational policies, strategic planning, annual evaluation and review of the centre's programs and budget within the framework of its approved five-year development plan and contributed through the meetings attended. Prior to her appointment at AKEPT, she was the UKM director of corporate communication from 2019-2020 and the UKM director/deputy director of International Relations Centre (UKM Global) from 2015-2019/ 2012-2015 respectively. Her extensive experience in internationalization of higher education leads to her appointment as an associate fellow at The National Higher Education Institute (IPPTN) Malaysia under the cluster of HEI Internationalization. She was attached to AirAsia Berhad in 2012 as an innovation and strategic analyst where she developed the IT strategic plan for AirAsia management, involved in various projects related to IT and innovation of AirAsia systems. Her landmark contribution to Malaysia's STEM development in Malaysia includes the establishment of the science and technology foresight Malaysia 2050 emerging science, engineering and technology (ESET) Study during her appointment as the Associate Fellow at the Academy of Sciences Malaysia (ASM) (2015-2019). Her research works in the area of Service Science is established via her writings in a book "Service Excellence for Sustainability: Lessons from Malaysia, Japan and Taiwan" and also in her postgraduate supervisions at the faculty. She can be contacted at email: yazrina@ukm.edu.my.